

CS663 - Digital Image Processing

Homework Assignment - 1

Omkar Shirpure (22B0910)
Aditya Saran (22B1839)
Suryansh Patidar (22B1036)

August 23, 2024



1

First Scenario: The difference in pixel sizes (0.5×0.5 vs. 0.25×0.25) suggests a uniform scaling factor in both directions (2x in both x and y directions). Affine transformations can handle scaling, rotation, translation, and shearing. So, an affine transformation would be a good motion model to align the two images.

Second Scenario: Here, the pixel sizes differ in the x and y directions (0.5×0.5 vs. 0.25×0.5), indicating non-uniform scaling. Affine transformations can still accommodate this difference by applying different scaling factors along the x and y axes. As with the previous case, affine transformations are sufficient to align the images since they can model translation, rotation, and scaling, which are all that's needed in this case.

2

I_1 , I_2 , and I_3 are related to each other by translational motion. It is defined that:

$$I_j = u_{ij}I_i \quad \forall \quad i, j \in \{1, 2, 3\}$$

We can write:

$$I_3 = u_{13}I_1 = u_{23}I_2 = u_{23}u_{12}I_1$$

$$\Rightarrow u_{13} = u_{23}u_{12}$$

During motion estimation, the above relationship holds true because any affine transformation can be decomposed into a series of more basic transformations, represented as the product of matrices. In this context, the motion from I_1 to I_3 can be expressed as a single transformation matrix or as a sequence of two transformations: first from I_1 to I_2 , and then from I_2 to I_3 .

3

Looking at the coordinates from graph and MATLAB both, we can see that a point corresponding to 630 on y-axis, we get MATLAB coordinates as (135,1435) and looking at another point of value 600 on y-axis, the MATLAB coordinates are (135,969).

The difference of MATLAB's coordinates is (0,466), whereas on graph it is only 30, so there can be scaling. So for calculating the transformation matrix, we will take 3 points and assume 6 variables corresponding to affine transformation matrix and will get 6 equations to calculate all the variable of transformation matrix

4

Dividing Equation 1 by f and Equation 2 by F , we get:

$$\frac{x_2}{f} = \frac{a}{f}x_1^2 + \frac{b}{f}y_1^2 + \frac{c}{f}x_1y_1 + \frac{d}{f}x_1 + \frac{e}{f}y_1 + 1$$

$$\frac{y_2}{F} = \frac{A}{F}x_1^2 + \frac{B}{F}y_1^2 + \frac{C}{F}x_1y_1 + \frac{D}{F}x_1 + \frac{E}{F}y_1 + 1$$

Transforming into a matrix equation, we can write:

$$\frac{\mathbf{x}_2}{f} = [\mathbf{x}_1 \quad \mathbf{y}_1 \quad 1] \begin{bmatrix} a/f & c/f & d/f \\ 0 & b/f & e/f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \\ 1 \end{bmatrix}$$

$$\frac{\mathbf{y}_2}{F} = [\mathbf{x}_1 \quad \mathbf{y}_1 \quad 1] \begin{bmatrix} A/F & C/F & D/F \\ 0 & B/F & E/F \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \\ 1 \end{bmatrix}$$

The transformation matrix consists of different affine transformations: shearing with $\frac{c}{f}$ and $\frac{C}{F}$, translation with $\frac{d}{f}, \frac{e}{f}$ and $\frac{D}{F}, \frac{E}{F}$, and scaling with $\frac{a}{f}, \frac{b}{f}$ and $\frac{A}{F}, \frac{B}{F}$. We can transform from x_1, y_1 to x_2, y_2 using these affine transformation matrices.

5

5.a

The MATLAB code for this part can be found in ***q5.m***

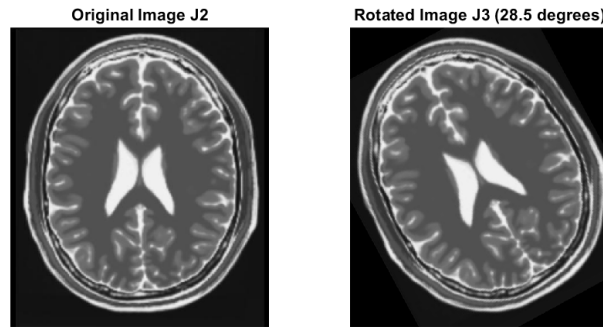


Figure 1: J2 vs after rotation J2

5.b

The MATLAB code for this part can be found in ***q5.m***

5.c

The plots of NCC, JE, QMI values versus θ .

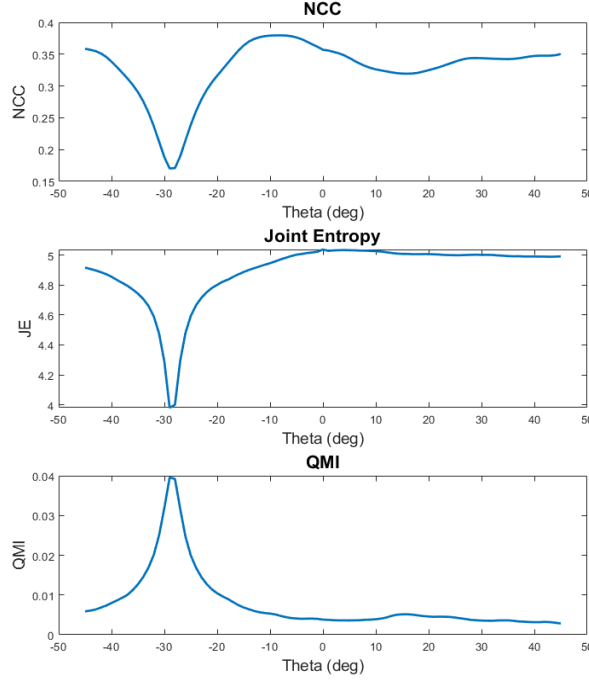


Figure 2: Alignment comparisons with θ using NCC, Joint Entropy, QMI

5.d

The optimal angles obtained using different methods were as follows (negative values indicate anti-clockwise rotation):

- Optimal angle based on NCC: -29.0 degrees
- Optimal angle based on JE: -29.0 degrees
- Optimal angle based on QMI: -29.0 degrees

The values across all three metrics suggest that the optimal alignment between the images is achieved at a rotation of -29.0 degrees.

The minima in the NCC and JE plots indicate that, at this angle, the images are most correlated and have the least joint uncertainty, respectively. Also, the QMI plot reaches its maximum at -29.0 degrees, showing that this rotation yields the highest amount of shared information between the images.

Overall, the alignment results from the joint analysis reinforce the accuracy of the identified rotation angle of -29 degrees, providing confidence in the robustness of these metrics for determining optimal image alignment.

5.e

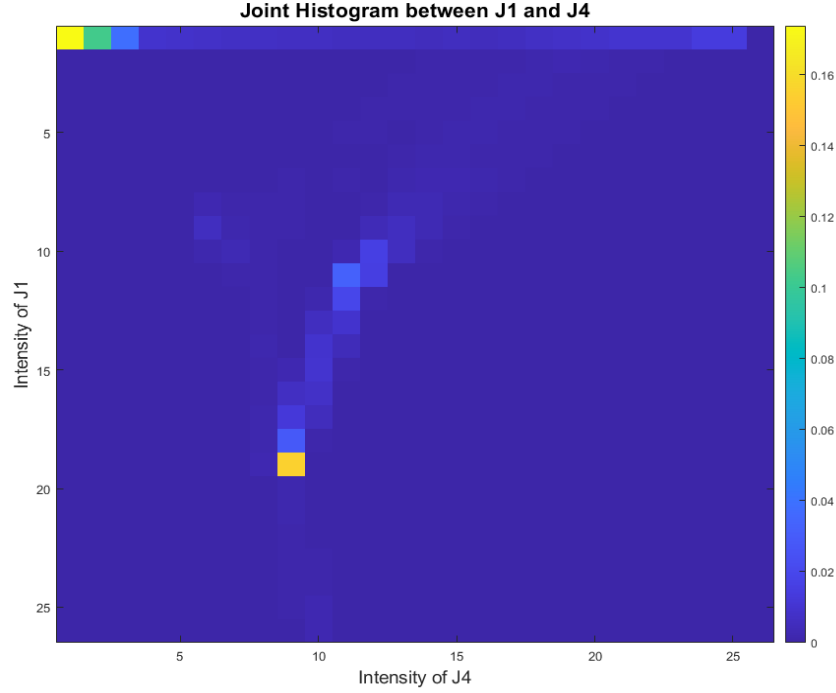


Figure 3: Joint Intensity histogram between J1 and J4

5.f

The main intuition behind Quadratic Mutual Information (QMI) is to measure the shared information between the intensity distributions of two images. QMI captures how dependent the images are on each other.

When the images are well-aligned, their shared information is maximized, resulting in a higher QMI value. Statistically, if the images were dependent, they share a significant amount of mutual information, leading to a higher QMI. Conversely, if the images were independent, their joint distribution would equal the product of their marginal distributions, resulting in a QMI value close to zero.

$$P(I_1, I_2) = P(I_1) \cdot P(I_2)$$

$$QMI = \sum_{i,j} (P(I_1, I_2) - P(I_1) \cdot P(I_2))^2$$

If I_1 and I_2 are independent, the term inside the summation is zero, and thus $QMI = 0$.

In alignment tasks, maximizing QMI leads to the rotation angle at which the images share the most mutual information, leading to better alignment. In summary, QMI also serves as a robust measure for determining optimal image alignment. By identifying the angle (but not limited to) that maximizes QMI, we can achieve more accurate image alignment, particularly in cases where complex dependencies between the images are present.

6

6.a

Points selection in images in MATLAB code.

6.b

The affine transformation matrix that we get from our code after choosing 12 pairs of points is,

$$\begin{pmatrix} 1.0244 & -0.0088 & 36.8330 \\ -0.0263 & 0.9887 & 25.5556 \\ 0 & 0 & 1.0000 \end{pmatrix}$$

6.c

Below is three images '*goi1.jpg*', '*goi2.jpg*' and the *warped image* after performing nearest neighbour interpolation.



Figure 4: Nearest neighbour Interpolation

6.d

Below is three images '*goi1.jpg*', '*goi2.jpg*' and the *warped image* after performing bilinear interpolation.



Figure 5: Bilinear Interpolation

6.e

If all the n points are collinear, using any 2 points, I can write a third point and i would not have sufficient points to make a sufficient number of equations that would calculate all the unknowns of the transformation matrix, and thus we will not be able to compute our transformation matrix.